Natural Language Processing

Lab 1: Introduction to Natural Language Processing

Objective

To understand what Natural Language Processing (NLP) is and why and where it is used. Introduction to Python, regular expressions.

Rules

Non-compliance with the following rules will result in serious consequences.

1. Students who are more than 15 minutes late are not allowed to sit in the lab.
2. Monitors will remain off until prompted.
3. Complete silence in the lab.
4. Do not speak in between lectures.
5. For any queries, stay silent and raise your hand.
6. Phones shut off or on silent during lab.
7. Labs will be submitted in hard copy.
8. The previous lab will be submitted in the first 15 minutes of the lab.
9. No late submissions will be accepted.
10. Treat lab tasks as your quiz or lab assignment.
11. No usage of A.I. for lab assignments or lab work.

Introduction

What?

1. A sub-field of Computer Science.
2. Generally associated with artificial intelligence.
3. Involves processing of natural language information.
4. Allows computers to perform tasks related to human language.

Why?

Three major reasons.

1. An enormous amount of information is now available in machine readable form as natural language text (newspapers, web pages, medical records, financial filings, etc.)

2. Conversational agents are becoming an important form of human-computer communication.
3. Much of human-human interaction is now mediated by computers via social media.

Where?

Three important applications.

1. Text analytics.
2. Question answering.
3. Machine translation.

Python and Regular Expressions

Python is an open source interpreted language. It is mostly used in web development and artificial intelligence related applications.

Regular expressions are a sequence of characters that specifies a match pattern in text.

Regular Expression Syntax

Regular expressions can contain both ordinary and special characters. Following is a list of some common special expressions used in Python.

- .
- *
- +
- ^
- $
- ?
- {m}
- {m, n}
- []
- |
- \d
- \D
- \s

- \S
- \w
- \W

Usage

```
import re

sent = “This is a sentence. This sentence is an example string.”
print(re.findall(“is”))
```

Common Functions

re.search

```
import re

sent = "This is a sentence. This sentence is an example string."
print(re.search(r'is', sent))
```

re.split

```
import re

sent = "This is a sentence. This sentence is an example string."
print(re.split(r'\s', sent))
```

re.findall

```
import re

sent = "This is a sentence. This sentence is an example string."
print(re.findall(r'sent', sent))
```

re.sub

```
import re

sent = "This is a sentence. This sentence is an example string."
print(re.sub(r'replacement', sent))
```

Lab Task 1

a) Redact all phone numbers in a document, replacing them with [REDACTED].
b) Reformate dates from MM/DD/YYYY to YYYY-MM-DD.
c) Clean up a string by removing all extra whitespace (replace multiple spaces/tabs/newlines with a single space).

Lab Task 2

Find all sequences of digits that are not part of a price (e.g., match 42 in "I have 42 apples", but not $42 or 42.99).